

Appendix

A Experimental details

A.1 Language modeling experiments

Here we provide details of the language modeling experiments presented in Sec. 4.1 and 4.3.

Training details. Table 5 shows the training and model hyper-parameters used to train the 340M and 1.3B parameter models. We train with an effective batch size of 64 per GPU with a sequence length of 2048, using 4 GPUs, for 28,672 steps; this yields 15,032,385,536 tokens. The 1.3B models were trained with a slightly increased sequence length of 2240; this increases the training token count to 16,441,671,680 (for simplicity, in the main text, we refer to both of them as trained for 15B tokens) Note that the actual parameter counts of 340M models are about 370M; we follow this convention from prior work [23, 6, 22, 8] (likely related to the fact that if the models shared the input and output embedding matrices of about 30M parameters, they would have 340M parameters; but this is not the case, either in our work, nor in the prior work).

Training of 340M models using 4 H100-80GB GPUs take about 8 hours for the baseline transformer and 10 hours for DeltaNet and all the HQLT models with the window size of 64 tokens. For the 1.3 models, these numbers become 26 hours for the baseline transformer and DeltaNet, and 30 hours for all the HQLTs variants.

We use the `fla` [17] toolkit to implement the models, and `flame` [18] to train them.

Table 5: Hyper-parameters of language models.

	Model	
	340M	1.3B
Number of layers	24	
Feedforward block multiplier	4	
Total hidden size	1024	2048
Number of heads	8	16
Sequence length	2048	2240
Effective Batch size	64	
Learning rate	$1e^{-3}$	
Warmup steps	1024	
Minimum learning rate	0.1	
Max norm clipping	1.0	
Std. of weight initializers	0.02	

Implementation details. Both the Synchronous and Delayed-Stream variants of HQLT studied in this work can directly make use of flash-attention [16] and flash-linear-attention [17] implementations (without modifying the corresponding Triton kernels) for the quadratic and linear components of the models, respectively. For Delayed-Block HQLT, we wrote a custom Triton kernel to replace intra-attention in the DeltaNet implementation of [23] by an efficient softmax attention implementation [16], and modified the backward function accordingly. Note, however, that there is still room for optimization as the intra and inter-chunk operations are currently implemented in two separate kernels (they may potentially be fused into a single kernel for further speed optimization).

Evaluation Details. Here we provide further descriptions of the evaluation datasets used in our general language modeling and retrieval tasks.

The six zero-shot common sense reasoning tasks we used in Table 2 are as follows. LAMBADA (LMB.) [33] is a task of predicting the last word in a sentence following some context sentences. PiQA [34] and HellaSwag (Hella.) [35] evaluate models’ common sense knowledge (learned in weights) through question answering with multiple choices. WinoGrande [36] (Wino.), inspired by the Winograd Schema Challenge [55], is a set of pronoun-resolution problems. The ARC dataset [37] is a set of grade school-level questions about natural science, which is split into two subsets,

ARC-easy (ARC-e) and ARC-challenge (Arc-c), according to their difficulties (determined based on whether certain baseline models can solve it). We use the standard `lm-evaluation-harness` [12] for evaluation on these datasets.

In Table 4, we use three additional datasets to evaluate models’ in-context retrieval abilities. SWDE is an information retrieval task based on the Structured Web Data Extraction dataset [56] (e.g., extracting some subject-predicate-object information from a raw HTML webpage about a movie). Similarly, FDA is an information retrieval task, extracting some key-value pairs from a set of PDFs from the FDA website. SQuAD [40] is the latest version of the Stanford Question Answering Dataset [57] which is a set of reading comprehension problems, which evaluate models’ ability to answer question based on a provided text passage. For SQuAD and SWDE, we use `lm-evaluation-harness` [12] for evaluation but by removing leading and trailing spacing in the document (see a related note at <https://github.com/EleutherAI/lm-evaluation-harness/issues/2690>). For FDA, we use the original script from https://github.com/HazyResearch/prefix-linear-attention/blob/main/lm-eval-harness/prompt_scripts/run_jrt_prompt_hf.sh (following the recommendation by Yang et al. [6] provided at <https://github.com/NVlabs/GatedDeltaNet?tab=readme-ov-file>).

The choice of all these tasks follow prior work [22, 23, 8].

A.2 Synthetic algorithmic tasks

Here we provide details about the experiments with two regular languages presented in Sec. 4.2. All the basic settings follow those of Grazzi et al. [8].

Tasks. In “Parity”, an input is a sequence of zeros and ones, and the task is to determine whether the number of ones in the sequence is odd or even. This essentially corresponds to modulo 2 addition, with the chance-level accuracy of 50%.

In “Modular Arithmetic (Mod Arith)”, the task is a modulo 5 addition and multiplication task (without brackets). Each symbol in an input sequence is either a number (from 0 to 4, in the module 5 case, which is our setting), or a mathematical symbol (there are five of those: $\{+, -, *, =, \text{eos}\}$ where the last symbol is the extra “end-of-sequence” token; which is not necessary for the modulo 5 case but included by convention). Here not only the output, but also the numbers in the sequences, are drawn between 0 to 4. This makes the total vocabulary size of 10 with a chance level accuracy of 20%.

Model configuration. The model hidden size is set to 128 and the number of heads is 4. For HQLTs, we use the dynamic vector mixing strategy, and the chunk size is set to 8 (except for the Delayed-Chunk variant, we set it to 16 for an implementation reason). We use 2 layers for Parity and 3 layers for Modular Arithmetic. Naturally, the crucial factor 2 is applied after the sigmoid activation on the dynamic learning rate β_t in DeltaNet (Eq. 8) to enhance its state-tracking ability [8].

Training settings. We train with a batch size of 1024 for 20,000 steps. We search for the best learning rate among $\{5e^{-3}, 1e^{-3}, 5e^{-4}, 1e^{-4}\}$ (the only difference compared to Grazzi et al. [8] is that this list includes $5e^{-3}$ instead of $1e^{-2}$), each with three seeds. We directly measure the validation accuracy to determine the best configuration. In the main result table, Table 3, we report the best/max result among the seeds for the best configuration, while Table 6 shows variability among seeds. (Note that we now have an updated Table 7 for the best results; we will replace Table 3 with this table in the final version. Our main conclusions are not affected by these updates). This is a standard practice in formal language recognition tasks [58, 9]. We train with sequence lengths from 3 to 40, and validate on sequences of lengths from 40 to 256.

Each training run on a single H100 takes about 70 min.

Evaluation. We report “normalized accuracies”, that is, by denoting the raw accuracy as A_{raw} and the chance level accuracy as A_{chance} , we report $(A_{\text{raw}} - A_{\text{chance}})/(100 - A_{\text{chance}})$, where A_{chance} is 50% for parity and 20% for modular arithmetic (modulo 5), such that all the accuracies are scaled to be between 0 and 100, where 0 is chance-level and 100 is perfect accuracy.

Table 6: **Median accuracy** and median absolute deviation over 3 seeds using the best learning rate for each case. This is to complete Table 3 which shows the best/max result among the seeds. The top-block results are taken from [8]. We show normalized accuracies [%] (0% is chance level).

Model	Parity acc ↑	Mod Arith acc ↑
Transformer [8]	0.3 ± 1.3	1.8 ± 0.9
Mamba [8]	100.0 ± 0.0	21.4 ± 2.7
DeltaNet [8]	99.9 ± 0.6	82.6 ± 14.6
HQLT		
Delayed-Stream	3.0 ± 0.3	22.2 ± 5.6
Delayed-Chunk	2.3 ± 0.1	1.4 ± 0.1
Synchronous	99.7 ± 0.1	93.2 ± 3.2
<i>w. Linear Attn.</i>	2.1 ± 0.3	32.9 ± 11.6

Table 7: (Updated Table 3 after complete hyper-parameter search) Evaluating Expressivity of Hybrid Quadratic-Linear Transformers (HQLTs) using regular language recognition tasks: Parity and Modular Arithmetic without brackets (Mod Arith). The top-block results are taken from [8]. We show normalized accuracies [%] (0% is chance level).

Model	Parity acc ↑	Mod Arith acc ↑
Transformer [8]	2.2	3.1
Mamba [8]	100.0	24.1
DeltaNet [8]	100.0	97.1
HQLT		
Delayed-Stream	3.3	27.8
Delayed-Chunk	2.8	1.4
Synchronous	100.0	97.0
<i>w. Linear Attn.</i>	2.5	44.5

B Further discussion

B.1 Practical computational costs of increasing the window size

In Sec. 4.3/Table 4, we demonstrated how increasing the quadratic attention window size in HQLTs improves their retrieval abilities. Here we discuss the computational costs for such improvements. In practice, within the range of window sizes discussed here, the actual computational cost is negligible, since during inference with a batch size of 1, short-window attention can be efficiently computed with a few/two steps of parallelizable matrix multiplication on a GPU, regardless of whether the window size is 64 or 1024. Training time is also only minimally affected by the increased window size as training is parallelized over the sequence elements. However, the total state size (which has a direct impact on the memory requirement) increases as a function of the window size S as $S(d_{\text{in}} + d_{\text{out}}) + d_{\text{out}} * d_{\text{in}}$ per layer and per head.

References

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, Long Beach, CA, USA, December 2017.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Int. Conf. on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

- 474 [3] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers
475 are RNNs: Fast autoregressive transformers with linear attention. In *Proc. Int. Conf. on Machine*
476 *Learning (ICML)*, Virtual only, July 2020.
- 477 [4] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic
478 recurrent networks. *Neural Computation*, 4(1):131–139, 1992.
- 479 [5] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear Transformers are secretly fast
480 weight programmers. In *Proc. Int. Conf. on Machine Learning (ICML)*, Virtual only, July 2021.
- 481 [6] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2
482 with delta rule. In *Int. Conf. on Learning Representations (ICLR)*, Vancouver, Canada, April
483 2025.
- 484 [7] Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo
485 Grazi. Deltaproduct: Improving state-tracking in linear RNNs via householder products.
486 *Preprint arXiv:2502.10297*, 2025.
- 487 [8] Riccardo Grazi, Julien Siems, Jörg KH Franke, Arber Zela, Frank Hutter, and Massimiliano
488 Pontil. Unlocking state-tracking in linear RNNs through negative eigenvalues. In *Int. Conf. on*
489 *Learning Representations (ICLR)*, Vancouver, Canada, April 2025.
- 490 [9] Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. Practical computational power of linear
491 transformers and their recurrent and self-referential extensions. In *Proc. Conf. on Empirical*
492 *Methods in Natural Language Processing (EMNLP)*, Sentosa, Singapore, 2023.
- 493 [10] James L McClelland, Bruce L McNaughton, and Randall C O’Reilly. Why there are comple-
494 mentary learning systems in the hippocampus and neocortex: insights from the successes and
495 failures of connectionist models of learning and memory. *Psychological review*, 102(3):419,
496 1995.
- 497 [11] Randall C O’Reilly and Kenneth A Norman. Hippocampal and neocortical contributions to
498 memory: Advances in the complementary learning systems framework. *Trends in cognitive*
499 *sciences*, 6(12):505–510, 2002.
- 500 [12] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles
501 Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas
502 Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron,
503 Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language
504 model evaluation harness, 07 2024. URL <https://zenodo.org/records/12608602>.
- 505 [13] Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell,
506 Colin A. Raffel, Leandro von Werra, and Thomas Wolf. The fineWeb datasets: Decanting
507 the web for the finest text data at scale. In *Proc. Advances in Neural Information Processing*
508 *Systems (NeurIPS)*, Vancouver, Canada, December 2024.
- 509 [14] Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou,
510 Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-
511 throughput tradeoff. In *Proc. Int. Conf. on Machine Learning (ICML)*, Vienna, Austria, July
512 2024. OpenReview.net.
- 513 [15] Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind:
514 Efficient infinite context transformers with infini-attention. *Preprint arXiv:2404.07143*, 2024.
- 515 [16] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning.
516 *Preprint arXiv:2307.08691*, 2023.
- 517 [17] Songlin Yang and Yu Zhang. Fla: A triton-based library for hardware-efficient implementa-
518 tions of linear attention mechanism, January 2024. URL [https://github.com/fla-org/](https://github.com/fla-org/flash-linear-attention)
519 [flash-linear-attention](https://github.com/fla-org/flash-linear-attention).
- 520 [18] Yu Zhang and Songlin Yang. Flame: Flash language modeling made easy, January 2025. URL
521 <https://github.com/fla-org/flame>.

- [19] Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 4331–4339, Barcelona, Spain, December 2016.
- [20] Mark A. Aizerman, Emmanuil M. Braverman, and Lev I. Rozonoer. Theoretical foundations of potential function method in pattern recognition. *Automation and Remote Control*, 25(6): 917–936, 1964.
- [21] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *Preprint arXiv:2307.08621*, 2023.
- [22] Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proc. Int. Conf. on Machine Learning (ICML)*, Vienna, Austria, July 2024.
- [23] Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, December 2024.
- [24] Imanol Schlag, Tsendsuren Munkhdalai, and Jürgen Schmidhuber. Learning associative inference using fast weight memory. In *Int. Conf. on Learning Representations (ICLR)*, Virtual only, May 2021.
- [25] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc V. Le. Transformer quality in linear time. In *Proc. Int. Conf. on Machine Learning (ICML)*, Baltimore, MA, USA, July 2022.
- [26] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216, 1990.
- [27] Tsendsuren Munkhdalai, Alessandro Sordoni, Tong Wang, and Adam Trischler. Metalearned neural memory. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 13310–13321, Vancouver, Canada, December 2019.
- [28] Bernard Widrow and Marcian E Hoff. Adaptive switching circuits. In *Proc. IRE WESCON Convention Record*, pages 96–104, Los Angeles, CA, USA, August 1960.
- [29] Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Going beyond linear transformers with recurrent fast weight programmers. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Virtual only, December 2021.
- [30] Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. A modern self-referential weight matrix that learns to modify itself. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 9660–9677, Baltimore, MA, USA, July 2022.
- [31] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *Preprint arXiv:2307.09288*, 2023.
- [32] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Int. Conf. on Learning Representations (ICLR)*, Toulon, France, April 2017.
- [33] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proc. Association for Computational Linguistics (ACL)*, Berlin, Germany, August 2016.
- [34] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *Proc. AAAI Conf. on Artificial Intelligence*, New York, NY, USA, February 2020.
- [35] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proc. Association for Computational Linguistics (ACL)*, Florence, Italy, August 2019.

- [36] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proc. AAAI Conf. on Artificial Intelligence*, New York, NY, USA, February 2020.
- [37] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *Preprint arXiv:1803.05457*, 2018.
- [38] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avani Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. Language models enable simple systems for generating structured views of heterogeneous data lakes. *Preprint arXiv:2304.09433*, 2023.
- [39] Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. Openceres: When open information extraction meets the semi-structured web. In *Proc. North American Chapter of the Association for Computational Linguistics on Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, June 2019.
- [40] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don’t know: Unanswerable questions for SQuAD. In *Proc. Association for Computational Linguistics (ACL)*, Melbourne, Australia, July 2018.
- [41] Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *Preprint arXiv:2402.19427*, 2024.
- [42] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proc. Int. Conf. on Machine Learning (ICML)*, Vienna, Austria, July 2024.
- [43] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, December 2024.
- [44] Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. An empirical study of mamba-based language models. *Preprint arXiv:2406.07887*, 2024.
- [45] Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *Preprint arXiv:2406.07522*, 2024.
- [46] Luca Zancato, Arjun Seshadri, Yonatan Dukler, Aditya Golatkar, Yantao Shen, Benjamin Bowman, Matthew Trager, Alessandro Achille, and Stefano Soatto. B’MOJO: Hybrid state space realizations of foundation models with eidetic and fading memory. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, December 2024.
- [47] Elvis Nunez, Luca Zancato, Benjamin Bowman, Aditya Golatkar, Wei Xia, and Stefano Soatto. Expansion span: Combining fading memory and retrieval in hybrid state space models. *Preprint arXiv:2412.13328*, 2024.
- [48] Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, et al. Hymba: A hybrid-head architecture for small language models. *Preprint arXiv:2411.13676*, 2024.
- [49] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.
- [50] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

- 617 [51] Zhen Qin, Songlin Yang, and Yiran Zhong. Hierarchically gated recurrent neural network for
618 sequence modeling. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*,
619 New Orleans, LA, USA, December 2023.
- 620 [52] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural
621 networks. In *Int. Conf. on Learning Representations (ICLR)*, Toulon, France, April 2017.
- 622 [53] Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. Simple recurrent units for highly
623 parallelizable recurrence. In *Proc. Conf. on Empirical Methods in Natural Language Processing*
624 (*EMNLP*), pages 4470–4481, Brussels, Belgium, November 2018.
- 625 [54] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in
626 partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.
- 627 [55] Hector J. Levesque. The winograd schema challenge. In *Logical Formalizations of Common-*
628 *sense Reasoning, AAAI Spring Symposium, Technical Report SS-11-06*, Stanford, CA, USA,
629 March 2011.
- 630 [56] Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: a unified solution
631 for structured web data extraction. In *Proc. of International Conference on Research and*
632 *Development in Information Retrieval (SIGIR)*, Beijing, China, July 2011.
- 633 [57] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+
634 questions for machine comprehension of text. In *Proc. Conf. on Empirical Methods in Natural*
635 *Language Processing (EMNLP)*, Austin, TX, USA, November 2016.
- 636 [58] Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transform-
637 ers to recognize formal languages. In *Proc. Conf. on Empirical Methods in Natural Language*
638 *Processing (EMNLP)*, pages 7096–7116, Virtual only, November 2020.